

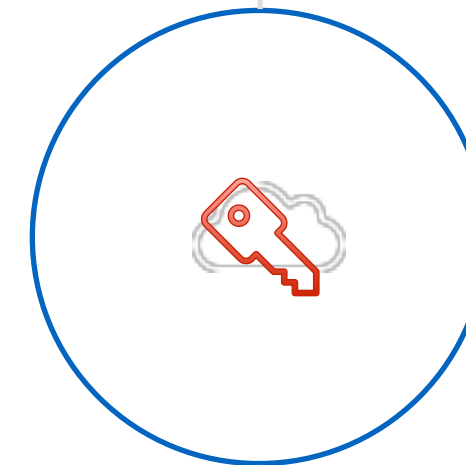
SPRINT5

目的はなにか

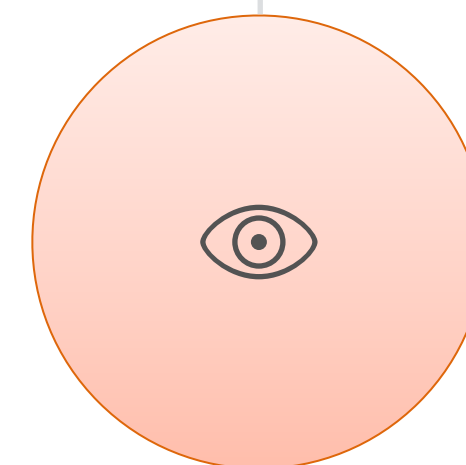


統計モデルを知る

スクラッチを通して**SVM**を理解する



線形モデルと異なる手法に触れる





このスライドは？

ここでは、SVMの
基本的な知識を学びましょう

SVMとはなにか

分類器としてのSVMは、入力に対し、直接**クラスラベル**を返すことから**識別関数**と呼ばれる。これに対し、ロジスティック回帰やニューラルネットワーク（分類器）の場合は、入力に対しクラスの条件付き**確率**（ $P(C = y_i | \mathbf{x}_i; \theta)$ ）を返す。クラスの条件付き確率（事後確率）をモデル化することから、これらを**分類モデル**と呼ぶ。分類モデルは、しきい値の設け方によってどのクラスに属するかを調整することができる。



この課題の対象者

- ① scikit-learnの分類モデルを用いて、学習、推定するコードが書ける方
- ② 分類モデルの計算過程（仮定関数、目的関数、最急降下法）を知っている方⁽¹⁾

(1) sprint4 ロジスティック回帰スクラッチを解いた方



この後の流れ

SVMの問題設定を知る

- ① 予測値を導く式（仮定関数）をたてる
- ② 最小化する問題を設定し、式（目的関数）をたてる
- ③ 主問題の目的関数を双対問題の目的関数に置き換える
- ④ 目的関数の最適解を探索的に求める（勾配法）
- ⑤ 得られた疎な解（サポートベクトル）を用いて推定する

Iris data

いまここにIrisデータセットがあるとしよう。

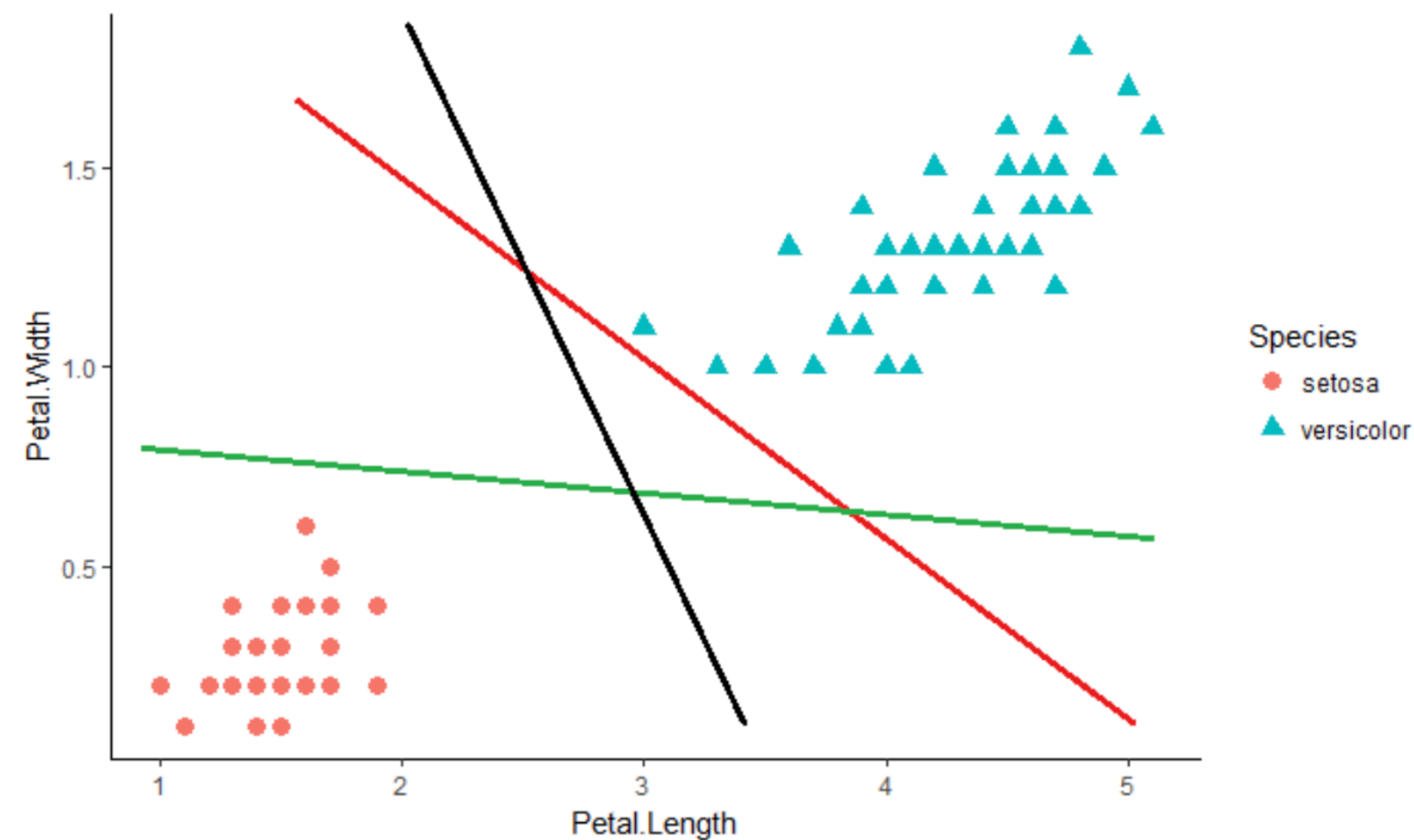
データ点はあらかじめクラスごとに色分けされている。

ある特徴量 X_1 (petallength) と特徴量 X_2 (petalwidth) を選び、

二変数間の関係 をプロットしてみよう。

今回もIris-setosaとIris-versicolorのクラスを分類するための識別境界（境界線）が引けると嬉しいが、SVMはロジスティック回帰のように**すべてのデータ点に対して**クラスに帰属する確率を出力しない。

どのようにして識別境界を引くのだろうか。





この後の流れ

SVMの問題設定を知る

- ① 予測値を導く式（仮定関数）をたてる
- ② 最小化する問題を設定し、式（目的関数）をたてる
- ③ 主問題の目的関数を双対問題の目的関数に置き換える
- ④ 目的関数の最適解を探索的に求める（勾配法）
- ⑤ 得られた疎な解（サポートベクトル）を用いて推定する

今回の仮定関数

今回も識別境界を引くために次のような仮定関数を想定しよう。

符号関数（sign）に線形結合を入力した出力：

線形結合： $w^T x$

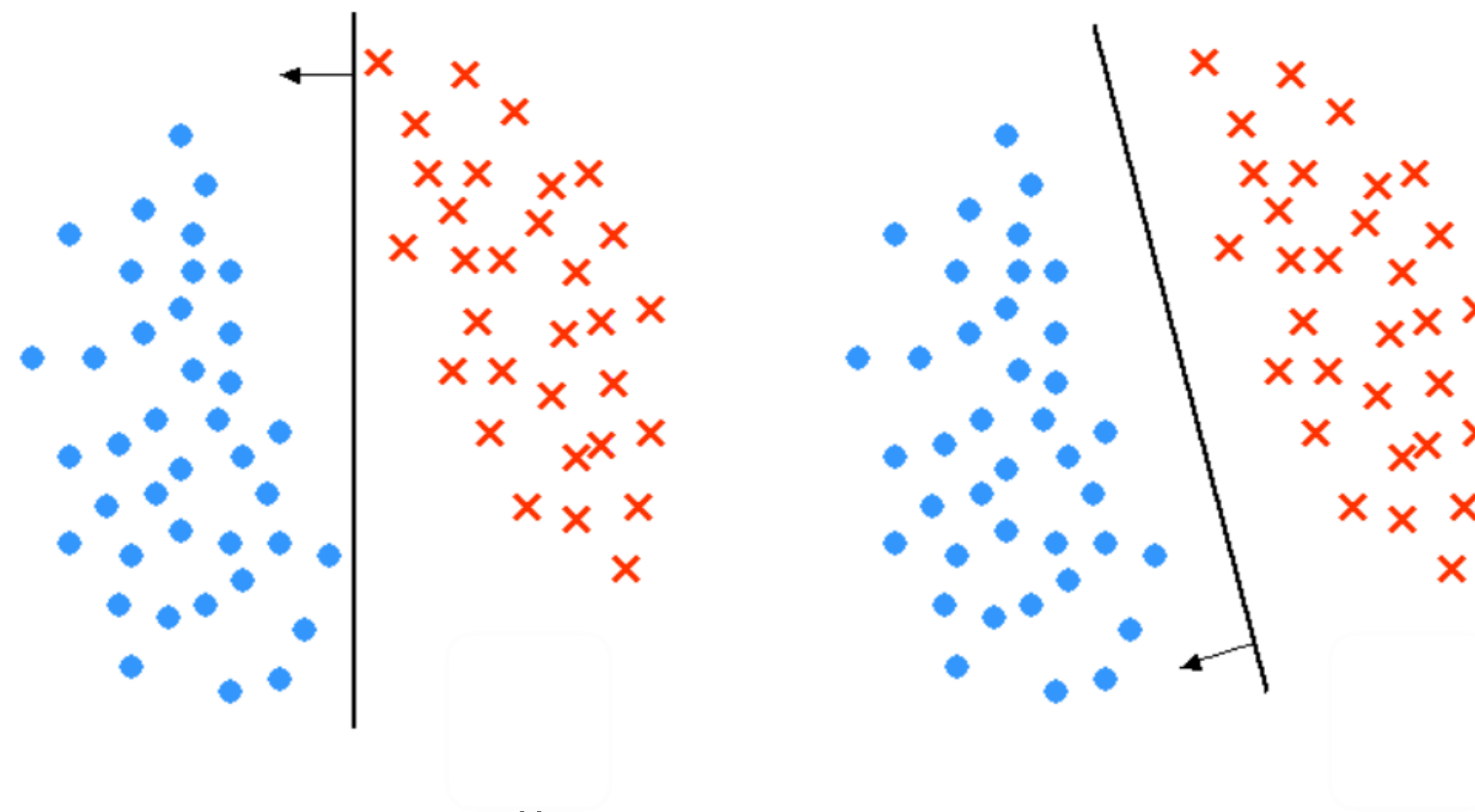
$$y = \text{sign}(w^T x)$$

sign関数は次のような出力を返す。

$$\text{sign}(w^T x) = \begin{cases} +1 & \text{if } (w^T x) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

$y \in \{-1, 1\}$ であるから、条件はまとめて $y (w^T x) \geq 0$ と書くことができる。

※ のちの双対問題を解き、疎な解（サポートベクトル）を得てはじめてこの仮定関数にて推定が可能になる。



$$y \in \{+1(\bullet), -1(\times)\}$$

識別境界からの距離

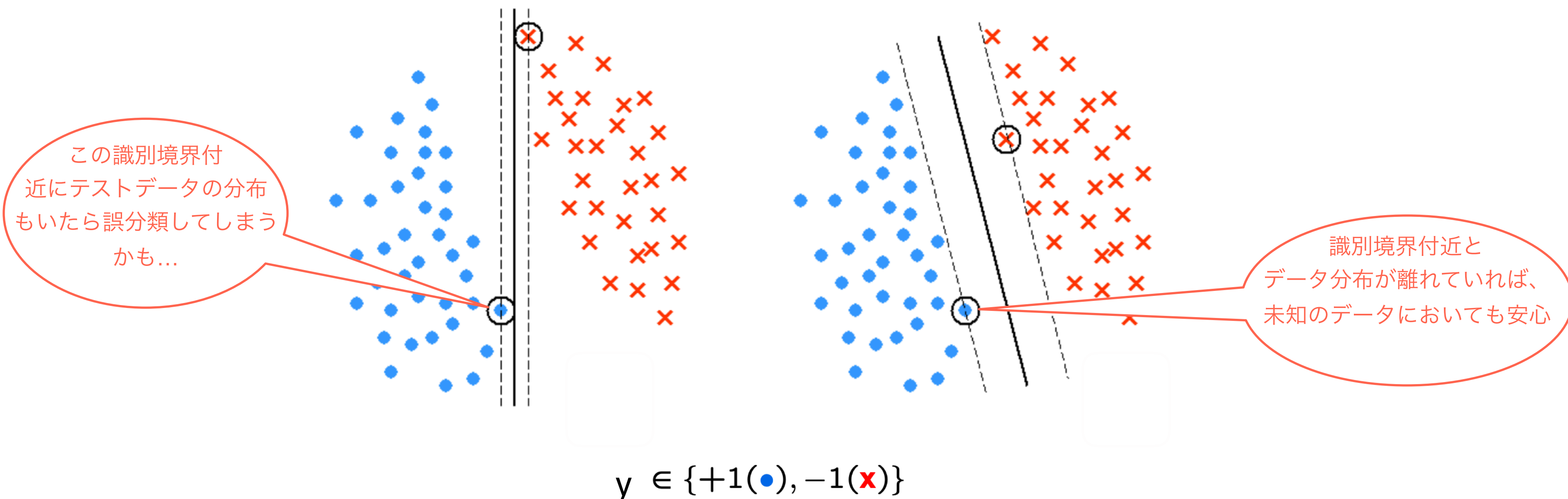
先ほどの仮定関数を用い、1と-1のクラス領域に分けるとい
う前提から出発しよう。それでは、どのようにパラメータを
制御し、識別境界の位置を決めるのが良いだろうか。

この問いへの答えは、**もっとも近いサンプルとの距離**（マー
ジン）が**最大**となるような識別境界を求めることである。

したがって、識別境界からの距離を計算する必要がある。
識別境界を $y(\mathbf{w}^T \mathbf{x}) = 0$ とすると、識別境界から各サンプル
(\mathbf{x}_i) までの距離は、 $\frac{y_i(\mathbf{w}^T \mathbf{x}_i)}{\|\mathbf{w}\|}$ で与えられる⁽¹⁾。

また、すべての i ($i \in N$) について $y_i(\mathbf{w}^T \mathbf{x}_i) \geq 0$ が成り立つ。

(1) 「点と線の距離の公式」より。

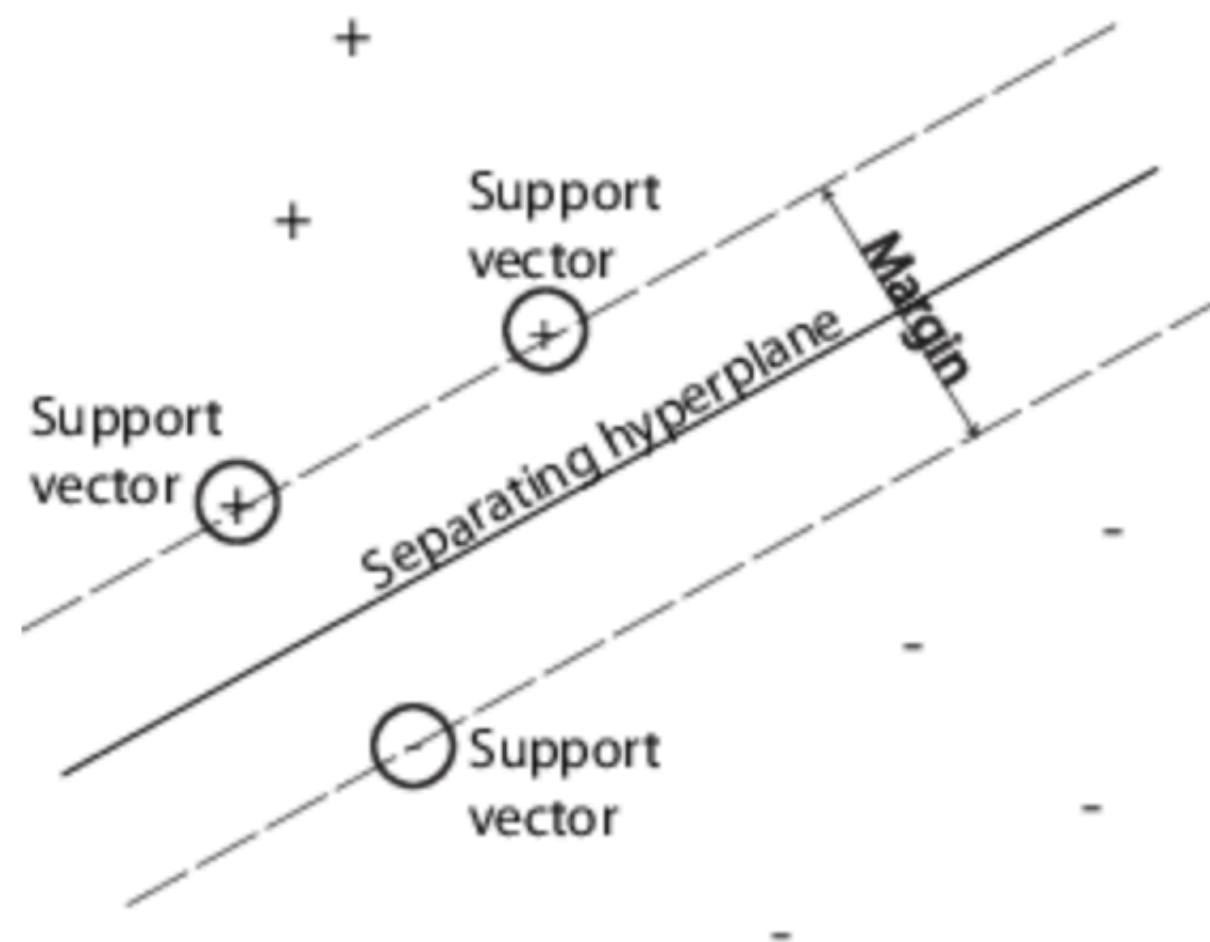


マージン最大化を目指す

マージンを最大化する前に、まず、**識別境界からの距離が最小である** x_i を求めたい。式で表すと以下のようになる。

$$\min_i \frac{y_i (w^T x_i)}{\|w\|}$$

このような x_i を**サポートベクトル**と呼ぶ。



w は i には依存しないため、 $\frac{1}{\|w\|}$ を \min_i の前に出す。

$$\frac{1}{\|w\|} \min_i (y_i (w^T x_i))$$

さらに、そのような x_i について、距離（式全体）が最大になるような w をを見つけるには、以下の式を最適化すればよい。

$$\arg \max_w \left\{ \frac{1}{\|w\|} \min_i (y_i (w^T x_i)) \right\}$$

その x_i から識別境界がもっとも遠ざかる
(マージンが最大になる) ような w を見つける

距離が最小になる i (つまり
 x_i : サポートベクトル) を求める

ここで、すべてのパラメータ(ベクトル w で表したすべての値)を、 kw というように定数倍した w' でも最適解になる(分母子ともに w は存在するゆえ、 w が k 倍されても相殺される)ため、識別境界からの最小の(x_i までの)距離が1となるように定数倍してもよい。

$$y_i (w'^T x_i) = 1$$

$$\arg \max_{w'} \left\{ \frac{1}{\|w'\|} \min_i (y_i (w'^T x_i)) \right\}$$



この後の流れ

SVMの問題設定を知る

- ① 予測値を導く式（仮定関数）をたてる
- ② 最小化する問題を設定し、式（目的関数）をたてる
- ③ 主問題の目的関数を双対問題の目的関数に置き換える
- ④ 目的関数の最適解を探索的に求める（勾配法）
- ⑤ 得られた疎な解（サポートベクトル）を用いて推定する

目的関数を立てる

$y_i (\mathbf{w}'^T \mathbf{x}_i) = 1$ とスケーリングすると $\min_i (y_i (\mathbf{w}'^T \mathbf{x}_i))$ が消え、さらに変数名 $\mathbf{w}' = \mathbf{w}$ とおきかえると、以下を最大化する問題に帰着する。

$$\arg \max_{\mathbf{w}} \frac{1}{\|\mathbf{w}\|}$$

これは $\|\mathbf{w}\|^{-1}$ を最大化する問題であるが、 $\|\mathbf{w}\|^2$ を最小化する問題として再定式化する。

$$\arg \min_{\mathbf{w}} \|\mathbf{w}\|^2$$

また、後で行う微分を容易にするため係数 $1/2$ をつけておく。

$$\arg \min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2$$

上で、 $y_i (\mathbf{w}'^T \mathbf{x}_i) = 1$ とスケーリングしたことから、すべての i ($i \in N$) について以下の制約条件が成立する。

$$y_i (\mathbf{w}^T \mathbf{x}_i) \geq 1$$

ここまでで成立した**目的関数**と**制約条件**を記す。

$$\begin{aligned} & \arg \min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to } y_i (\mathbf{w}^T \mathbf{x}_i) \geq 1 \end{aligned}$$

このような線形不等式あるいは線形等式による制約条件において二次関数を最小化するような問題を二次計画問題と呼ぶ。

二次計画問題は、局所解が最適解になることが保証されており、同じく二次計画問題である**ラグランジュ未定乗数法 (KKT条件つき)** を解くことによって、最適解が得られる。これは双対定理⁽¹⁾に基づく問題の置き換えである。

上の目的関数と、不等式制約条件を組み合わせた L (**ラグランジュ関数**) を以下のように定義する。

$$L(\mathbf{w}, \lambda) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \lambda_i \{y_i (\mathbf{w}^T \mathbf{x}_i) - 1\}$$

式中の λ (ラムダ) は**ラグランジュ乗数**と呼ばれる。

この式中の \mathbf{w} を λ を用いた式で置き換えると、最適化すべき (主問題に対する) 双対表現が得られる (スライド p20 を参照)。主問題が目的関数の**最小化を目的**とする場合、双対問題ではラグランジュ関数を**最大化する**ことを目的とする。

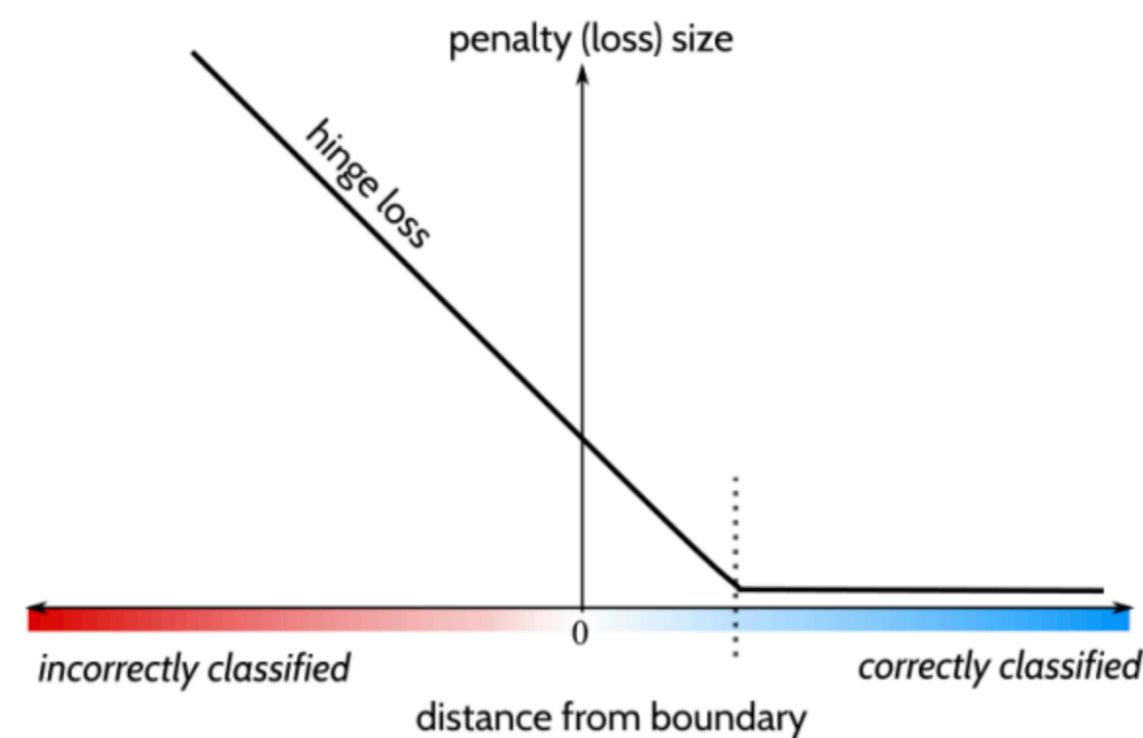
(1) 主問題と双対問題のいずれか一方が最適解を持つなら、もう一方も最適解を持ち、主問題の最小値と双対問題の最大値は一致する。

<https://ja.wikipedia.org/wiki/%E5%8F%8C%E5%AF%BE%E5%95%8F%E9%A1%8C>

制約条件について

前出したSVMの制約条件 $y_i (w^T x_i) \geq 1$ は、損失関数として定式化できる。これを**ヒンジ損失関数**という。SVMのヒンジ損失関数は、 $y_i (w^T x_i)$ が1になるまで線形に減少し、1以上では出力が0で一定になる関数。後ろに正則化項がつくこともある。

ヒンジ損失関数

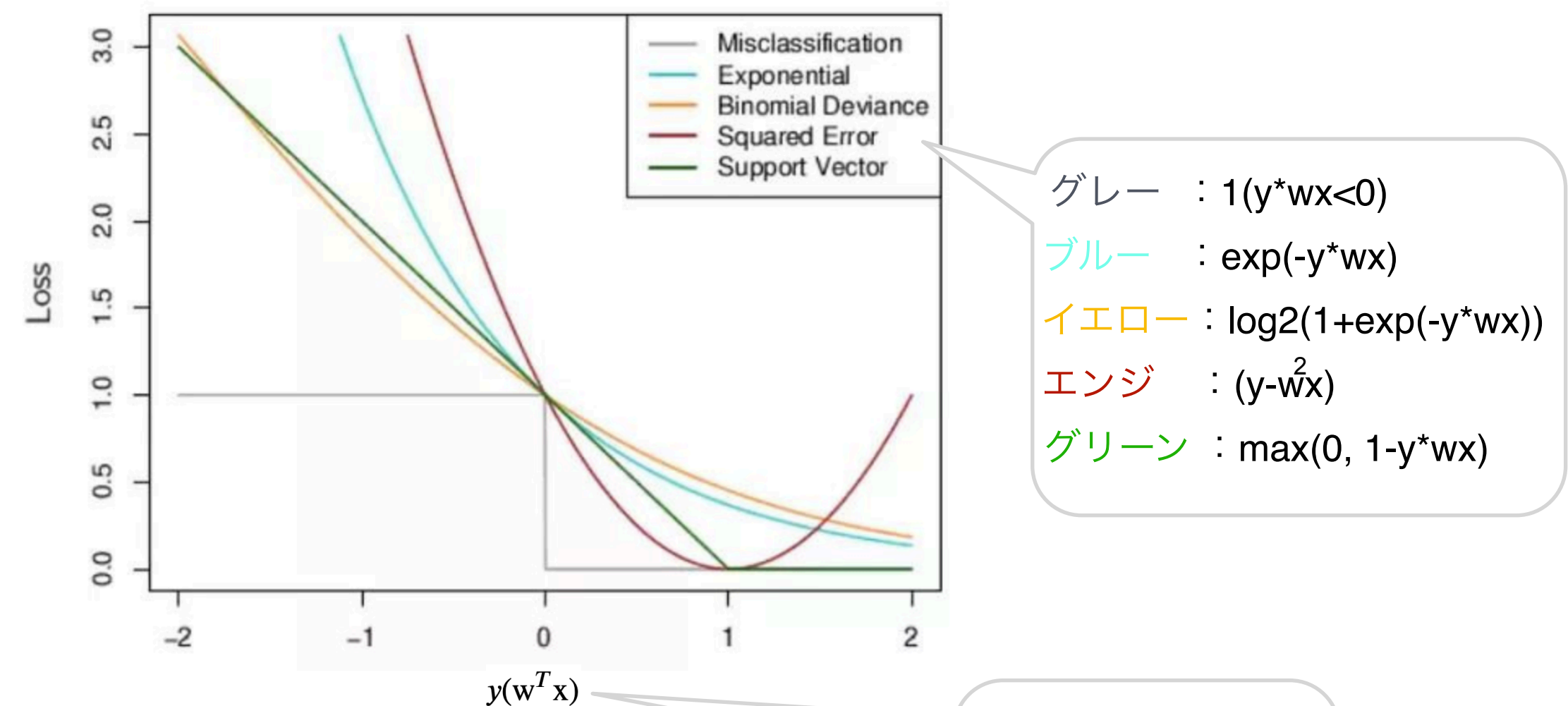


ヒンジ損失関数

$$\text{hinge}((w^T x), y) = \max\{0, 1 - y(w^T x)\}$$

$y(w^T x)$ が1以上の（識別境界から遠ざかる）とき、損失のペナルティを与えない。代わりに平均二乗誤差を用いると、十分に識別されているサンプルにもペナルティを与えるのに対し、ヒンジ損失関数の場合、サポートベクトルより遠く（ $y(w^T x) \geq 1$ ）にいるサンプルには影響を与えない。一方、サポートベクトルより識別境界に近いほど大きなペナルティを課される。

損失関数の比較



<https://rohanvarma.me/Loss-Functions/>

SVMのヒンジ損失関数のパラメータ



この後の流れ

SVMの問題設定を知る

- ① 予測値を導く式（仮定関数）をたてる
- ② 最小化する問題を設定し、式（目的関数）をたてる
- ③ 主問題の目的関数を双対問題の目的関数に置き換える
- ④ 目的関数の最適解を探索的に求める（勾配法）
- ⑤ 得られた疎な解（サポートベクトル）を用いて推定する

ラグランジュ未定乗数法

これは、制約条件の下で関数の極値を求める**等号制約付き最適化手法**である。具体的には、 $g(x) = 0$ という**制約条件**の下において、 $f(x)$ という**目的関数**を最小（あるいは最大）にする x を求める。以下の式は最適化問題が満たすべき条件の一般形式である。

$f(x)$ は最小化したいが
 $L(x, \lambda)$ 全体は最大化する

$$\frac{\partial L}{\partial x} = 0$$

$$\frac{\partial L}{\partial \lambda} = 0$$

$$L(x, \lambda) = f(x) - \lambda^T g(x)$$

$$\lambda = (\lambda_1, \dots, \lambda_m)^T, g(x) = (g_1(x), g_2(x), \dots, g_m(x))^T$$

ラグランジュ関数の一般形式

今回は $f(x)$ を最小にする問題だが、
最大化する問題の場合は

$$L(x, \lambda) = f(x) + \lambda^T g(x)$$

$1/2 \|w\|^2$ は最小化したいが
 $L(x, \lambda)$ 全体は最大化する

$$L(w, \lambda) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \lambda_i (y_i (w^T x_i) - 1)$$

今回のラグランジュ関数 (p15参照)

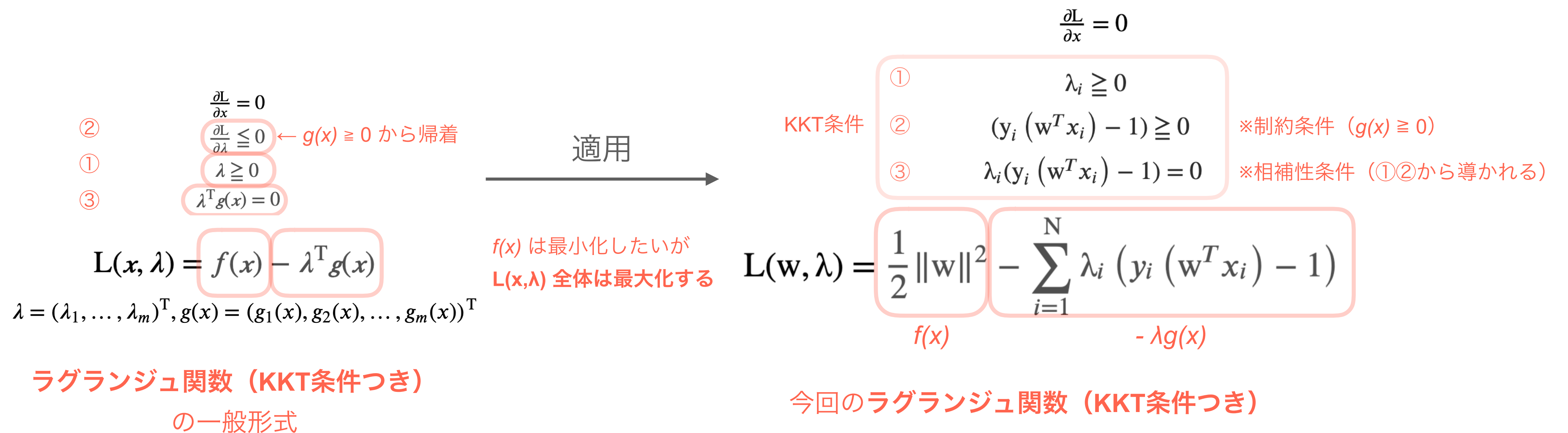
主問題の目的関数と制約条件を
このラグランジュ関数の形式に
落としこむ

KKT条件 (カルーシュ・クーン・タッカー条件)

最適化問題を解く際に極値が満たすべき必要条件。ラグランジュ未定乗数法にKKT条件を適用し解法を求めることの利点は、ラグランジュ未定乗数法が一般化され、等号のみならず**不等号制約**も扱える点である。

SVMの目的関数は**不等号制約** $y_i (w^T x_i) \geq 1$ (p15,16参照) を持つため、**ラグランジュ未定乗数法 (KKT条件つき)** にて定式化し双対問題を解く。

形式的には、 **$g(x) \geq 0$** という**不等号制約条件**にて、 **$f(x)$** という**目的関数**を最小化するxを求める。



双対表現と勾配法の準備

前述のラグランジュ関数を式展開し、**双対表現**を得る。

さらに、双対表現をラグランジュ乗数にて微分し、後の勾配法のための**勾配**を準備する。

式展開の途中で、SVMのパラメータである w について式を微分して得られる $w = \sum_{i=1}^N \lambda_i y_i x_i$ を代入し、ラグランジュ関数のパラメータをラグランジュ乗数のみとする。

(w 以外にバイアス表現(b)が含まれる場合も、同様に消去できる)

$$\begin{aligned}
 L(w, \lambda) &= \overset{f(x)}{\frac{1}{2} \|w\|^2} - \overset{-\lambda g(x)}{\sum_{i=1}^N \lambda_i (y_i (w^T x_i) - 1)} \\
 &= \frac{w^T w}{2} - \sum_{i=1}^N \lambda_i (y_i (w^T x_i) - 1) \\
 &= \frac{1}{2} \left(\sum_{i=1}^N \lambda_i y_i x_i \right)^T \left(\sum_{i=1}^N \lambda_i y_i x_i \right) - \sum_{i=1}^N \lambda_i \left\{ y_i \left(\left(\sum_{i=1}^N \lambda_i y_i x_i \right)^T x_i \right) - 1 \right\} \\
 &= \frac{1}{2} \sum_{i,j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j + \sum_{i=1}^N \lambda_i - \sum_{i=1}^N \lambda_i y_i \left(\sum_{i=1}^N \lambda_i y_i x_i \right)^T x_i \\
 &= \frac{1}{2} \sum_{i,j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j + \sum_{i=1}^N \lambda_i - \sum_{i,j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j \\
 &= \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i,j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j
 \end{aligned}$$

※ラグランジュ関数を w で微分して得られる $w = \sum_{i=1}^N \lambda_i y_i x_i$ を代入する

上式の目的関数を λ_j で微分すると、こちらの**勾配**を得る。

$$\frac{\partial L(\lambda)}{\partial \lambda_i} = 1 - y_i \sum_{j=1}^N \lambda_j y_j x_i^T x_j$$

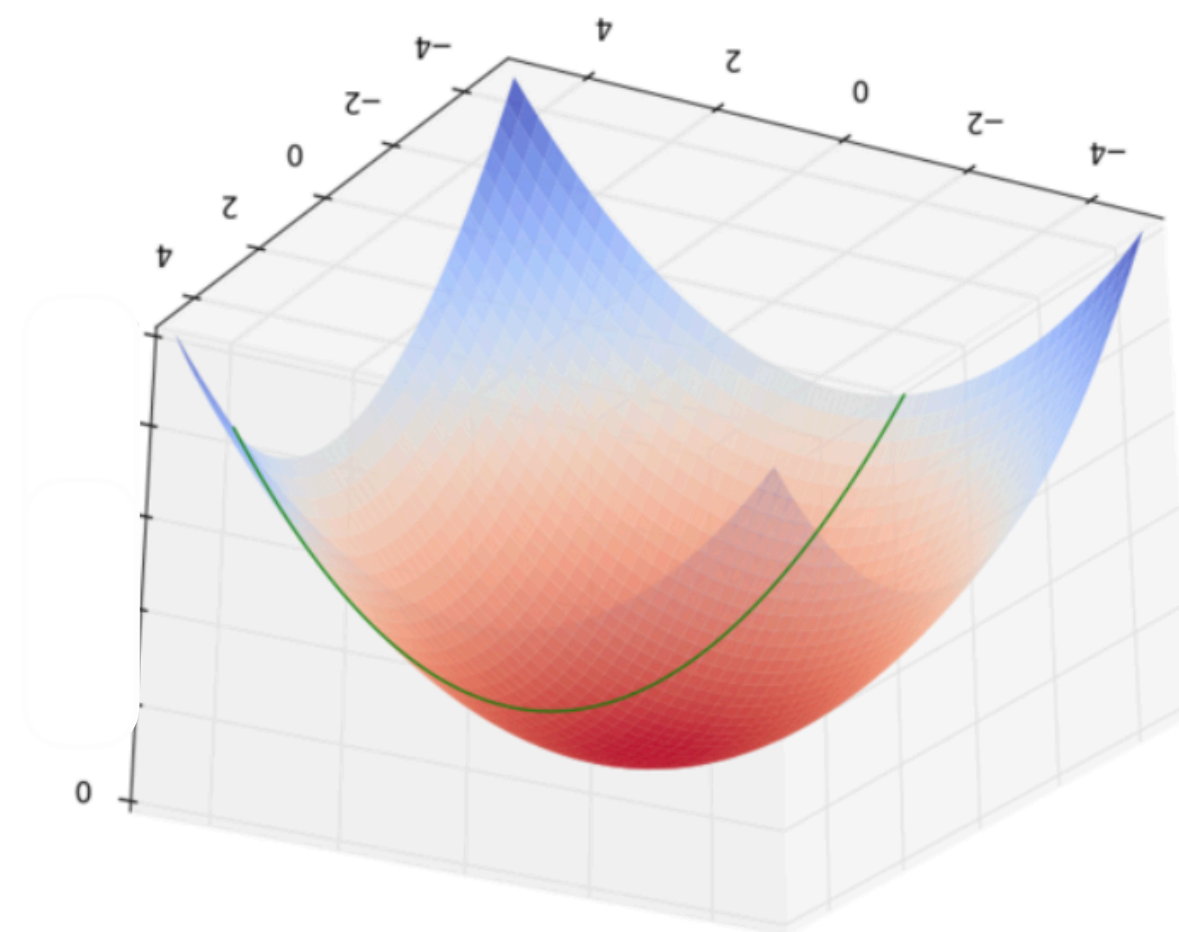
これを主問題 (SVM) に対する**双対表現**とする。(後式の Σ をまとめているがDiverの式と同じ目的関数)

ラグランジュ関数が示していること

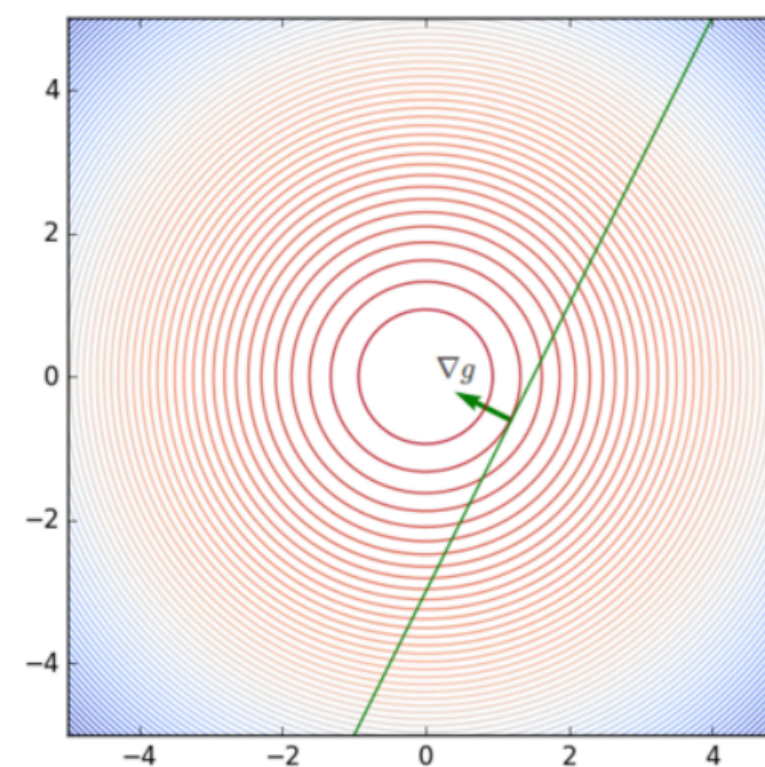
幾何学的関係を見て、直感的に理解してみよう。

$f(\mathbf{x})$ を下図のように下に凸の3次元曲面で表したとき、**曲面上の曲線（等高線）**は $f(\mathbf{x}) = z$ を表現している。

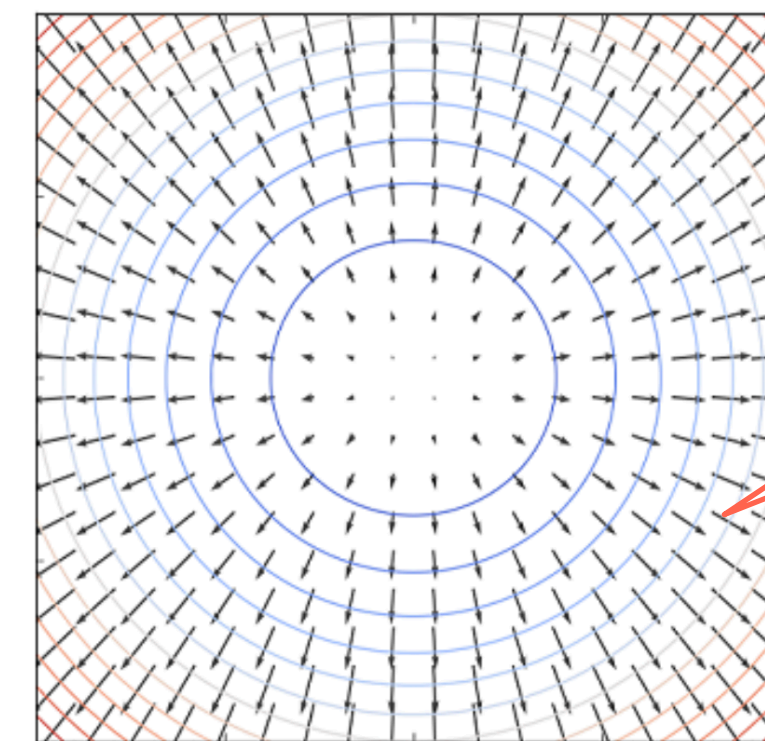
さらに、 $g(\mathbf{x}) = 0$ もまた**曲線（緑の線）**で描くことができる。



$f(\mathbf{x})$ と $g(\mathbf{x})=0$



$g(\mathbf{x}) = 0$ の法線ベクトル



等高線の法線ベクトル

等高線と $g(\mathbf{x}) = 0$ が接している点では、KKT条件より勾配ベクトル ∇f と ∇g は平行になる（今回はベクトルの向きも同じ）。曲面の勾配ベクトル ∇f とは等高線の法線ベクトルのことであり、この法線ベクトルは原点から放射線状に広がっている。



この後の流れ

SVMの問題設定を知る

- ① 予測値を導く式（仮定関数）をたてる
- ② 最小化する問題を設定し、式（目的関数）をたてる
- ③ 主問題の目的関数を双対問題の目的関数に置き換える
- ④ 目的関数の最適解を探索的に求める（勾配法）
- ⑤ 得られた疎な解（サポートベクトル）を用いて推定する

ラグランジュ関数の最大化

双対表現 (p20参照) を λ (ラグランジュ乗数) で微分して得た勾配で、 λ を更新する。

この双対表現の関数は、 λ に対して最大化を目指すため、 λ に勾配を加算して更新する。このように関数の最大値を探す場合は、勾配上昇法 (gradient ascent mothod) と呼ぶ。

$$\frac{\partial L(\lambda)}{\partial \lambda_i} = 1 - y_i \sum_{j=1}^N \lambda_j y_j x_i^T x_j$$
$$\lambda_i^{new} = \lambda_i + \alpha (1 - y_i \sum_{j=1}^N \lambda_j y_j x_i^T x_j)$$

勾配

学習率



この後の流れ

SVMの問題設定を知る

- ① 予測値を導く式（仮定関数）をたてる
- ② 最小化する問題を設定し、式（目的関数）をたてる
- ③ 主問題の目的関数を双対問題の目的関数に置き換える
- ④ 目的関数の最適解を探索的に求める（勾配法）
- ⑤ 得られた疎な解（サポートベクトル）を用いて推定する

仮定関数

最適解を求めた結果得られた疎な解（サポートベクトル： s_n ）と、仮定関数（p11参照） $f(x) = \text{sign}(w^T x)$ を用いて推定を行う。

仮定関数に現れる w は $w = \sum_{i=1}^N \lambda_i y_i x_i$ によって置き換えられる。

sign関数は、 $\text{sign}(w^T x) = \begin{cases} +1 & \text{if } (w^T x) \geq 0 \\ -1 & \text{otherwise} \end{cases}$ より、1か-1の値を返す。

$$f(x) = \text{sign}(w^T x) \\ = y(w^T x)$$

この等式はラグランジュ関数を w で微分すると得られる。（p20参照）

ここで w を $w = \sum_{i=1}^N \lambda_i y_i x_i$ にて置き換える

推定用の仮定関数：

$$f(x) = \sum_{n=1}^N \lambda_n y_n x^T s_n$$

未知のデータ

サポートベクトル

サポートベクトルの数



SVMの問題設定を知った

- ① 予測値を導く式（仮定関数）をたてる
- ② 最小化する問題を設定し、式（目的関数）をたてる
- ③ 主問題の目的関数を双対問題の目的関数に置き換える
- ④ 目的関数の最適解を探索的に求める（勾配法）
- ⑤ 得られた疎な解（サポートベクトル）を用いて推定する